CERTIFICATE OF ELECTRONIC TRANSMISSION

I hereby certify that this correspondence for Application No. 10/758,484 is being electronically transmitted to Technology Center 2186, via EFS-WEB, on March 2, 2009.

/Scott A. Stinebruner/   _March 2, 2009_____

Scott A. Stinebruner, Reg. No. 38,323   Date

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte Kevin Curtis Griffin, Scott Dennis Helt,

Michael James McDermott, Glen W. Nelson and Mark Philip Piazza

Appeal No. _____

Application No: 10/758,484

Applicant:	Kevin Curtis Griffin et al.	Art Unit:	2186
Application No.:	10/758,484	Examiner:	Shane M. Thomas
Filed:	January 15, 2004		
For:	ASYNCHRONOUS HYBRID MIRRORING SYSTEM		

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

RESPONSE TO NOTIFICATION OF NON-COMPLIANT BRIEF

Dear Sir or Madam:

This is in response to the Notification of Non-Compliant Appeal Brief mailed February 9, 2009. This Response is timely filed. As set forth below, Section V of the Appeal Brief, titled "Summary of the Claimed Subject Matter," has been revised to contain every limitation of independent claim 14. In addition, Appellants have fixed the typographical error located on page 5, line 3 of the Appeal Brief. Please replace Section V of the originally filed Appeal Brief with the revised Section V.

V. <u>SUMMARY OF CLAIMED SUBJECT MATTER</u>

Appellants' invention is generally directed towards methods for efficiently and reliably mirroring data of a primary system to a backup system. In one respect, a method consistent with embodiments of the invention provides a hybrid approach that enjoys benefits of both the fast processing of asynchronous mirroring and the data integrity of synchronous mirroring. More particularly, a number of update requests are organized into groups at both the primary and backup systems based on update requests from an application such that a first request from the application creates a first group of update requests and a second request from the application creates a second group of update requests. *See Application*, page 8 lines 2-13, page 11 line 25 to page 12 line 5, and page 13 lines 1-20, as well as FIG. 4 blocks 64-76. Although the first and second groups of update requests have an order dependency relative to each other, the update requests in each of the first and second groups of update requests are capable of being processed concurrently and without regard for order relative to one another. *See Application*, page 5 lines 2-11, page 5 lines 12-17, and page 8 lines 2-13, page 11 line 25 to page 12 lines 5, page 14 lines 16-24, and page 15 lines 18-22, as well as FIG. 4 blocks 76-82, FIG. 5 blocks 86-92, FIG. 6 blocks 102-144, and FIG. 7 blocks 152-166. Thus, the respective groups of update requests are completed sequentially to preserve sequential ordering of those groups. The updates of each group, however, are completed concurrently, or substantially at the same time and without regard to order. *See Application*, Abstract and page 5 lines 2-11, page 5 lines 12-17 and page 8 lines 2-13, page 11 line 25 to page 12 line 5, page 13 line 1 to page 14 line 24, page 14 line 25-31 and page 15 line 1 to page 17 line 2, as well as FIG. 4 blocks 76-82, FIG. 5 blocks 86-92, FIG. 6 blocks 102-144, and FIG. 7 blocks 152-166. This allows improved processing times with regard to the requests of each group.

Despite recent rapid advances in technology, computing systems partially or completely lose their ability to function properly during a crash or failure, losing valuable mission critical data. *See Application*, page 1 lines 16-27. Once practice used to protect critical data involves data mirroring of the memory of a primary computer system to a backup computer system. Thus, write requests in the primary computer system are transmitted for execution to the backup computer system, and in the event that the primary computer system fails a user is switched to the backup computer system without a loss of significant amounts of data. *See Application*, page

1 line 28 to page 2 line 7. The prior art discloses two mirroring approaches for backing up data: the synchronous mirroring approach and the asynchronous mirroring approach. Synchronous mirroring involves updating data on the backup computer in the same order as the data updates on the primary computer system and preserves the order of the updates on the primary computer system. *See Application*, page 2 lines 8-15. However, synchronous mirroring often suffers from poor performance in that it takes a relatively long period of time for a particular update on the backup system to complete in proper order. *See Application*, page 2 line 25 to page 3 line 2. Thus, subsequent updates on the primary and backup systems are delayed and, over time, the ordered requirement of synchronous mirroring can result in unacceptable delays that adversely affect system performance. *See Application*, page 2 line 25 to page 3 line 2.

Asynchronous mirroring, on the other hand, provides improved performance over synchronous mirroring by issuing updates at the backup node without regard to order as the backup computer system attempts to process all received updates in parallel. *See Application*, page 3 lines 3-10. Thus, update requests are processed when received and while other requests are processing and the backup computer system puts the updates in proper sequential order after the application stops running. *See Application*, page 3 lines 3-10. However, this results in less integrity of the data on the backup computing system than asynchronous mirroring, as the primary computer system may crash and updates on the backup computer system may be processed out of order, rendering the data on the backup computer system out of order and unusable. *See Application*, page 3 lines 16-20.

Moreover, the inefficiency and unreliability of existing mirroring techniques becomes exacerbated in a clustered computer environment, as the nodes of a clustered computer environment process work in parallel. *See Application*, page 3 lines 21-30. As such, update requests for an application configured across the clustered computer environment may be completed simultaneously across a plurality of nodes, thus placing a larger burden on the clustered computing system to efficiently and accurately back up data from the plurality of nodes. *See Application*, page 3 lines 21-30.

Embodiments consistent with embodiments of the invention address these drawbacks by providing methods for updating data at a backup system that track updates to a primary system. Independent claim 1, for example, recites that, in response to receiving a first update request

from an application, a first group is created that includes a first plurality of update requests which further includes the first update request. In response to receiving a second update request from the application prior to completing the first plurality of update requests, a second group is created that includes a second plurality of update requests which further includes the second update request. The first update request of the first plurality of update requests in the first group has an order dependency relative to the second update request of the second plurality of update requests in the second group, with the update requests in each of the first and second groups capable of being processed concurrently and without regard for order relative to one another. The method further includes concurrently completing the first plurality of update requests of the first group, and, after concurrently completing the first plurality of update requests, concurrently completing the second plurality of update requests of the second group.

Independent claim 14, for example, recites synchronously processing a plurality of groups of update request in that a first update request from an application in a first group of update requests from among the plurality of groups has an order dependency relative to a second update request from the application in a second group of update requests from among the plurality of groups, with the update requests in each group being capable of being processed concurrently and without regard for order relative to one another, and wherein receipt of the second update request prior to processing of the first update request initiates the creation of the second group of update requests. The method further comprises asynchronously processing the update requests in reach group.

Claims 1 and 14 are independent claims, while the rest of claims 3, 4, 6-13, 15, 16, and 36 are dependent claims. Claims 2, 5, 17-35, and 37 are canceled. For the convenience of the Board, each of the independent claims has been reproduced below and annotated with references to the specification and drawings to satisfy the requirement to concisely explain the claimed subject matter:

1. A method for updating data at a backup system (page 5 lines 2-11, page 8 lines 2-13, and page 14 line 25 to page 17 line 2, as well as FIG. 6 and FIG. 7) that tracks updates made to a primary system (page 5 lines 2-11, page 8 lines 2-13, and page 12 line 22 to page 14 line 24, as well as FIG. 4 and FIG. 5), the method comprising:

in response to receiving a first update request from an application (page 12 lines 22-30 and page 14 lines 25-31, as well as FIG. 4 block 64 and FIG. 6 block 102), creating a first group (page 8 lines 2-13, page 11 line 25 to page 12 line 5, and page 13 lines 1-20, as well as FIG. 4 blocks 70-76) including a first plurality of update requests (page 8 lines 2-13, page 11 line 25 to page 12 line 5, page 12 line 22 to page 14 line 24, and page 14 line 25 to page 17 line 2, as well as FIG. 4 blocks 64-82 and FIG. 6 blocks 102-140), the first plurality of update requests including the first update request; (page 8 lines 2-13, page 11 line 25 to page 12 line 5, page 13 lines 1-20, and page 15 line 12 to page 17 line 2, as well as FIG. 4 blocks 70-76 and FIG. 5 blocks 116-144)

in response to receiving a second update request from the application prior to completing the first plurality of update requests (page 12 line 22 to page 13 line 20, page 14 lines 25-31, and page 15 line 12 to page 17 line 2, as well as FIG. 4 block 64 and 70-76, and FIG. 6 block 102 and 116-144), creating a second group (page 8 lines 2-13, page 11 line 25 to page 12 line 5, and page 13 lines 1-20, as well as FIG. 4 blocks 70-76) including a second plurality of update requests (page 8 lines 2-13, page 11 line 25 to page 12 line 5, page 12 line 22 to page 14 line 24, and page 14 line 25 to page 17 line 2, as well as FIG. 4 blocks 64-82 and FIG. 6 blocks 102-140), the second plurality of update requests including the second update request (page 8 lines 2-13, page 11 line 25 to page 12, line 5, page 13 lines 1-20, and page 15 line 12 to page 17 line 2, as well as FIG. 4 blocks 70-76 and FIG. 6 blocks 116-144), the first update request of the first plurality of update requests in the first group having an order dependency relative to the second update request of the second plurality of update requests in the second group (page 11 line 25 to page 12 line 5, page 13 lines 1-20, page 14 lines 16-24, and page 16 lines 6-29), with the update requests in each of the first and second groups capable of being processed concurrently and without regard for order relative to one another; (page 5 lines 2-11, page 5 lines 12-17, page 8 lines 2-13, page 11 line 25 to page 12 lines 5, page 14 lines 16-24, and page 15 lines 18-22, as well as FIG. 4 blocks 76-82, FIG. 5 blocks 86-92, FIG. 6 blocks 102-144, and FIG. 7 blocks 152-166) concurrently completing the first plurality of update requests of the first group; (page 5 lines 2-11, page 5 lines 12-17, page 8 lines 2-13, page 11 line 25 to page 12 line 5, page 13 line 1 to page 14 line 24, page 14 lines 25-31, and page 15 line 1 to page 17 line 2, as well as FIG. 4 blocks 76-82, FIG. 5 blocks 86-92, FIG. 6 blocks 102-144, and FIG. 7 blocks 152-166) and

after concurrently completing the first plurality of update requests, concurrently completing the second plurality of update requests of the second group. (page 5 lines 2-11, page 5 lines 12-17, page 8 lines 2-13, page 11 line 25 to page 12 line 5, page 13 line 1 to page 14 line 24, and page 14 line 25 to page 17 line 2, as well as FIG. 4 blocks 70-82, FIG. 5 blocks 86-92, FIG. 6 blocks 102-144, and FIG. 7 blocks 152-166).

14. A method for updating data at a backup system (page 5 lines 2-11, page 8 lines 2-13 and page 14 line 25 to page 17 line 2, as well as FIG. 6 and FIG. 7) that tracks updates made to a primary system (page 5 lines 2-11, page 8 lines 2-13, and page 12 line 22 to page 14 line 24, as well as FIG. 4 and FIG. 5) the method comprising:

synchronously processing a plurality of groups of update requests (page 5 lines 2-11, page 5 lines 12-17, page 8 lines 2-13, page 11 line 25 to page 12 line 5, page 13 line 1 to page 14 line 24, and page 14 line 25 to page 16 line 12, as well as FIG. 4 blocks 70-82, FIG. 6 block 102-144, and FIG. 7 blocks 152-166), a first update request from an application in a first group of update requests from among the plurality of groups having an order dependency relative to a second update request from the application in a second group of update requests from among the plurality of groups (page 11 line 25 to page 12 line 5, page 13 lines 1-20, page 14 lines 16-24 and page 15 line 12 to page 17 line 2), with the update requests in each group being capable of being processed concurrently and without regard for order relative to one another (page 5 lines 2-11, page 5 lines 12-17, page 8 lines 2-13, page 11 line 25 to page 12 line 5, page 12 line 22 to page 14 line 24, and page 15 lines 18-22, as well as FIG 4 blocks 64-74 and FIG. 5 blocks 86-92), and wherein receipt of the second update request prior to processing of the first update request initiates the creation of the second group of update requests (page 11 line 25 to page 12 line 5, page 12 line 22 to page 13 line 20, page 14 lines 25-31, and page 15 line 23 to page 16 line 12, as well FIG. 4 block 64 and 70-76, and FIG. 6 block 102 and 116-144); and

asynchronously processing the update requests in each group. (page 5 lines 2-11, page 5 lines 12-17, page 8 lines 2-13, page 11 line 25 to page 12 line 5, page 13 line 1 to page 14 line 24, page 14 lines 25-31, and page 15 line 1 to page 17 line 2, as well as FIG. 4 blocks 76-82, FIG. 5 blocks 86-92, FIG. 6 blocks 102-144, and FIG. 7 blocks 152-166).

Other features will be discussed in greater detail in the arguments section below. In addition, it should be noted that, as none of the claims recite any means plus function or step plus function elements, no identification of such elements is required pursuant to 37 CFR §41.37(c)(1)(v). Furthermore, there is no requirement in 37 CFR §41.37(c)(1)(v) to provide support for the subject matter in the separately argued dependent claims, as none of these claims recite means plus function or step plus function elements, and so no discussion of any of these claims is provided.

Conclusion

Appellants respectfully submit that the previously submitted Appeal Brief, with revised Section V above, is in full compliance with the provisions of 37 CFR § 41.37. Appellants have reviewed the previously filed Appeal Brief and believe there are no additional errors. If this Response leaves any issues open, a call to undersigned counsel at (513) 241-2324 would be gratefully appreciated. Moreover, if any other charges or credits are necessary to complete this communication, please apply them to Deposit Account 23-3000.

Respectfully submitted,

March 2, 2009

Date

  /Scott A. Stinebruner/

Scott A. Stinebruner
Reg. No. 38,323
WOOD, HERRON & EVANS, L.L.P.
2700 Carew Tower
441 Vine Street
Cincinnati, Ohio 45202
Telephone: (513) 241-2324
Facsimile: (513) 241-6234